International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 13, Issue 5, May-2025, Available online at: www.ijaresm.com

Campus Placements Prediction & Analysis using Machine Learning

Peram Rithika Reddy[1], Bathula Chandana Priya[2], Chennepalli Chaithanya Reddy[3], Balaraju Pavitra[4], Dr. Renuka Devi M[5]

[1,2,3,4]Student BCA, School of Information Science, Presidency University, Bangalore, Karnataka, India
[5]Professor, School of Information Science, Presidency University, Bangalore, Karnataka, India

----------**********----------

ABSTRACT

We have developed web-based Hiring Prediction application using Flask, simplifies the hiring process by assessing candidates based on academic and skill parameters. Users can securely register, log in, and participate in aptitude tests covering Logical Reasoning, Verbal Skills, Programming Skills, and Communication Skills. These tests evaluate candidates' problem-solving, technical, and interpersonal abilities. The system leverages a machine learning model to predict hiring eligibility based on criteria such as SSLC Percentage, PU Percentage, UG CGPA, internship status, backlog history, and test scores. Predictions follow predefined thresholds, ensuring fair evaluation. The platform is secure, using bcrypt for password hashing and session management for user protection. Regular users can view their prediction history, while administrators access a dashboard to manage user data and refine recruitment strategies. SQLite is used for efficient data storage, maintaining user credentials, test results, and contact messages. Flask's lightweight and scalable framework supports rapid development, while Python libraries like joblib ensure seamless integration of the pre-trained machine learning model. This data-driven solution not only streamlines recruitment for organizations but also empowers candidates by highlighting skill gaps and offering actionable insights for improvement.

Index Terms **-Hiring Prediction Application, Flask Framework, Machine Learning Model, Aptitude Tests Secure Platform, Data-Driven Recruitment.**

INTRODUCTION

The Hiring Prediction Web Application is a Flask-based platform that predicts hiring eligibility using machine learning. On the frontend, it offers a user-friendly interface designed with HTML and CSS, allowing users to register, log in, and take aptitude tests covering Logical Reasoning, Verbal, Programming, and Communication Skills. Users can enter academic scores like SSLC, PU, and UG CGPA, view prediction history, and access support through an intuitive layout. The backend, powered by Flask, handles user authentication, session management, and data processing. It uses SQLite for secure data storage, with bcrypt ensuring password hashing. Candidate eligibility is determined based on predefined criteria like SSLC ≥ 50%, PU ≥ 35%, UG CGPA > 5.0, and aptitude test scores like Logical, Verbal, Programming, Communication must each be ≥ 15. A pre-trained machine learning model analyzes user inputs and generates hiring predictions in real-time. Administrators access a dashboard to manage user data, monitor predictions, and refine strategies. This scalable, data-driven system automates recruitment, providing fair and consistent candidate evaluations.

1.1 Background

The Hiring Prediction Web Application is a platform developed using Flask in Python to help users predict their hiring eligibility. It uses machine learning to analyze user data, such as academic scores like SSLC, PU, UG CGPA, internship experience, backlog status, and aptitude test scores like Logical, Verbal, Programming, and Communication Skills. Users can securely register, log in, and take skill assessments. Their data is stored securely using SQLite, and passwords are protected with bcrypt. Administrators can access a dashboard to manage user data and monitor system performance. The application is user-friendly, with a clean interface built using HTML and CSS, and it provides fast predictions using a pre-trained machine learning model.

Problem statement

Student placement prediction based on student characteristics is a tedious process in the education sector.The general Placement Prediction System considers only academic performances in order to predict whether a student can be hired or not hired. Judging the student based only on his academic performances would be unfair for the student, since a student could be having good aptitude, technical and communication skills but unfortunately might not be good in academic performance.

Overview of the solution

To address the challenge of predicting student placement eligibility, we propose a solution that leverages machine learning to analyze a combination of academic metrics and interpersonal skills. The system is designed to accept user input in the form of student performance data like academic scores, technical skills, aptitude results, communication ability, etc., process the data, and then generate a real-time prediction indicating whether the student is likely to be hired or not hired.

The solution is implemented using a Python-based Flask web framework for the backend, HTML and CSS for the front end, and SQLite for secure data storage. A trained machine learning model is embedded into the system to make accurate predictions, and bcrypt is used for secure password hashing during user registration and login.

LITERATURE SURVEY

[1] Liu Yang et al. conducted the study "Database Management and Performance Analysis for Student Placement using ML," which analyzed student scores using a Decision Tree and psychological factors through Association Rules. The research classified student performance into "poor" and "excellent" categories. [ACM, 2018]

[2] Alfiani et al. conducted a study titled "Mapping Student Performance Patterns Using Data Mining," they used student scores to classify them into low, average, and high performers through K-means clustering. The research identified hidden patterns in student performance based on GPA, lecture grades, and lab grades. [Elsevier, 2019]

[3] Meena A. and Gopal M. conducted a study titled "Recruitment Prediction System," which employed Logistic Regression and Decision Tree models to predict placements based on academic and skill-based inputs. Logistic Regressionprovided better interpretability, while the Decision Tree offered % higher accuracy of 82%. [Springer, 2020]
[4] Rajesh T. and Swetha G. conducted a study titled "Employee Hiring Using Machine Learning," where they utilized the Random Forest algorithm on labeled candidate resume data after preprocessing and feature selection. The model achieved an accuracy of 85%. [IJERT, 2021]

[5] Arvind K. and Sneha J. conducted the study "Predictive Analytics for Hiring," which analyzed historical hiring data using Support Vector Machine (SVM) and KNN models. The study found that SVM was more precise in handling large datasets, achieving 88% accuracy compared to KNN's 80%. [IEEE, 2022]

[6] Sheetal M. B and Savita Bakare conducted research titled "Campus Placement Prediction Using Machine Learning," which compared the accuracy and execution time of KNN and Fuzzy Logic models. The findings showed that KNN outperformed Fuzzy Logic in prediction reliability and speed. [Springer, 2022]

[7] Sumitha R. et al. conducted the study "Predicting Student Placement Outcomes Using Decision Trees," which compared three models J48 Decision Tree, Naive Bayes, and Multilayer Perceptron. The J48 Decision Tree achieved the highest accuracy, outperforming the other models. [IEEE, 2021]

METHODOLOGY

Data Collection

We started by gathering Candidate data from various historical records, including candidate academic scores, internship experience, backlogs status, and aptitude scores from aptitude tests assessing Logical Reasoning, Verbal Skills, Programming Skills, and Communication Skills and stored in the Csv file as dataset.csv. Data was collected through various sources, preprocessed to ensure consistency, and consolidated into a unified dataset.

Data Preprocessing

Handling Missing Values: Missing values were identified and managed using appropriate imputation techniques, such as mean or mode imputation for numeric and categorical values.

Categorical Encoding: Encoded categorical variables, for example internship experience, backlog status using Label Encoding for efficient model training.

Data Splitting: Divided the dataset into training and testing sets with an 80-20 ratio, ensuring the model is trained on most data while being evaluated on unseen data.

Feature Scaling: Numeric features, for example SSLC Percentage, PU Percentage, UG CGPA were normalized using Min-Max Scaling to maintain consistent scales across features.

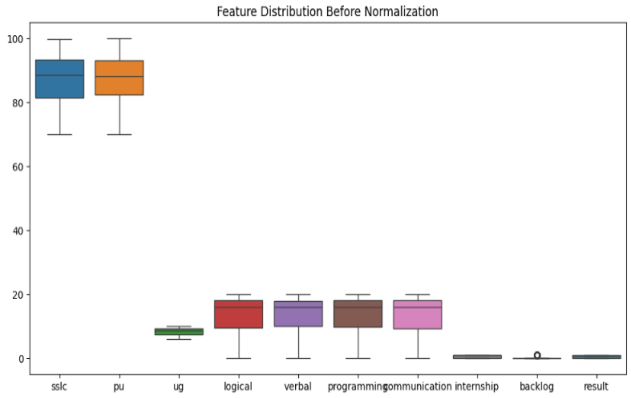


Fig 1 - Feature Distribution Before Normalization

The above graph shows feature distributions before normalization, with SSLC and PU percentages having significantly higher scales than other attributes. This imbalance can lead to biased model performance. Normalization is essential to ensure all features contribute equally. Outliers in SSLC and PU are also noticeable.

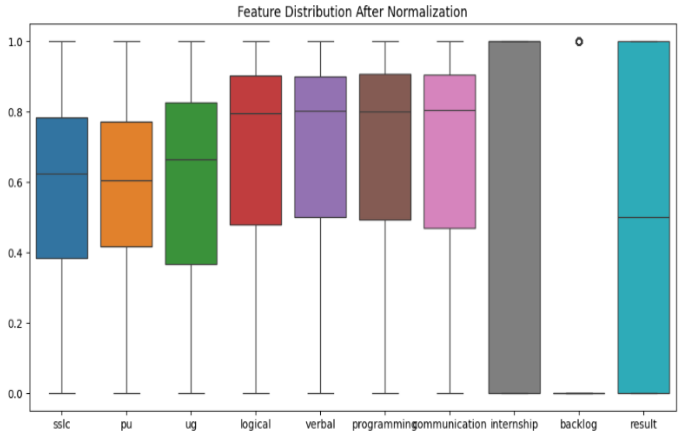


Fig 2 - Feature Distribution After Normalization.

The above box plot represents the normalized feature distribution for the dataset. The normalization process scales all features to a similar range, typically 0 to 1 to remove bias caused by differing scales. Each feature's spread and median are centered on the same range, allowing for fair comparisons and improved performance of algorithms sensitive to feature scaling, such as gradient descent-based models.

Exploratory Data Analysis (EDA and Visualizations):

Before training the model, we performed EDA to understand patterns, relationships, and potential anomalies in the dataset using graphs like:

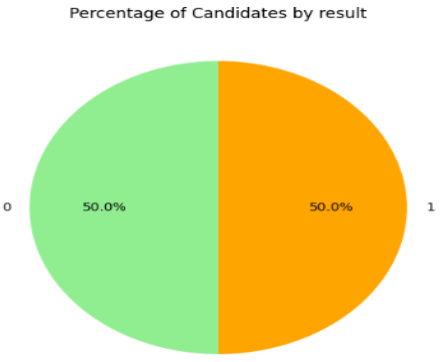


Fig 3 - The proportion of hired(1) vs not hired(0).

In the above pie chart, the results are evenly distributed between hired (1) and not hired (0), each accounting for 50% of the data. This balance ensures an unbiased dataset, providing a fair evaluation for the machine learning model.

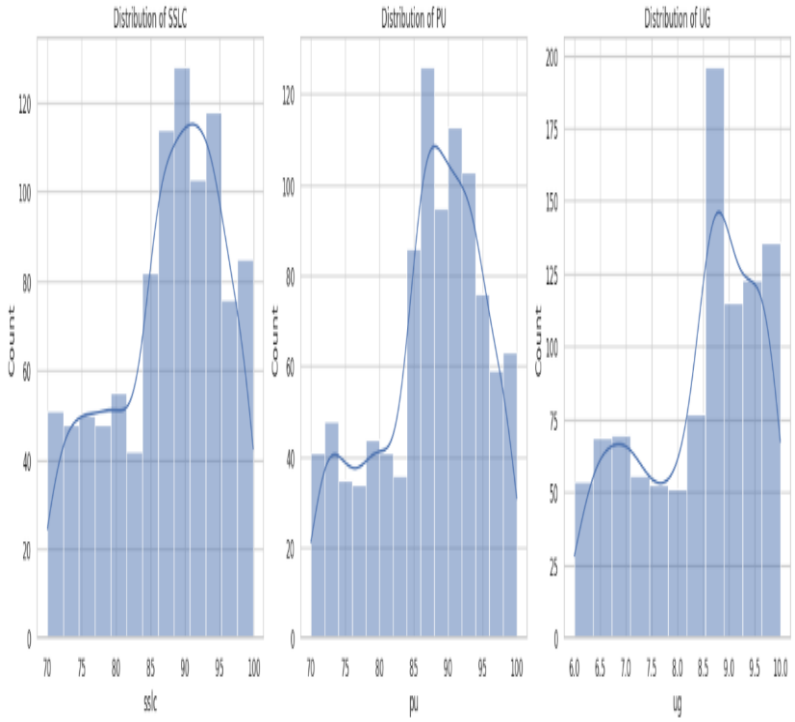


Fig 4 – Distributions of SSLC, PU, and UG Percentage

In the above graph, the distributions of SSLC, PU, and UG academic scores indicate a relatively high concentration of candidates scoring between 85 and 95 in SSLC and PU percentages. For UG CGPA, scores are more varied but still show a concentration towards the higher end i.e. around 8.5 to 9.5. This suggests that most candidates have strong academic performance in their education.

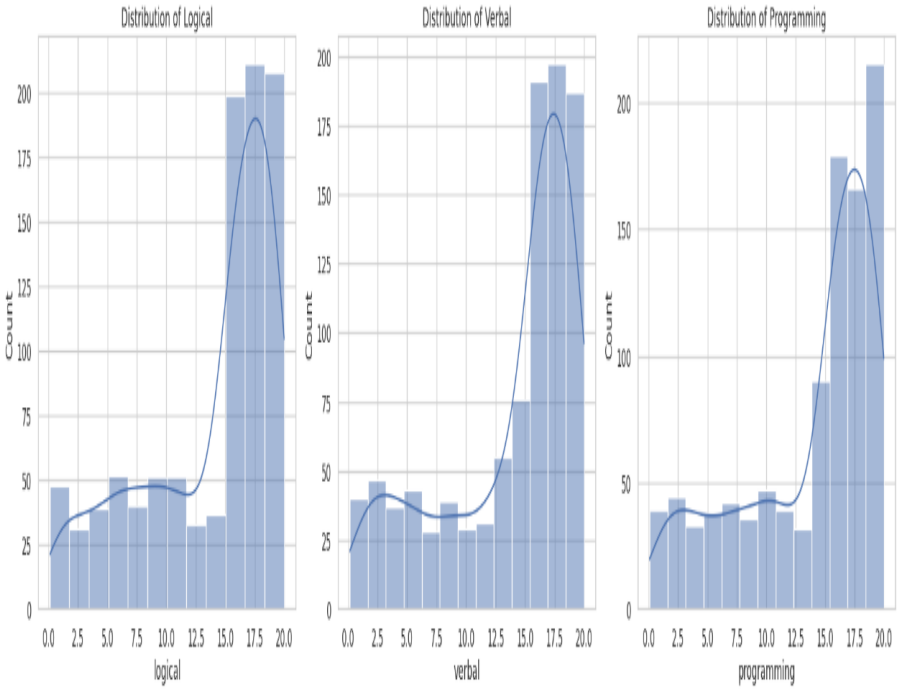


Fig 5 - Distributions of Logical, Verbal, and Programming skills

From the above graph, it can be concluded that the distribution plots for Logical, Verbal, and Programming scores reveal that most candidates score in the higher range of 15-20 in these aptitude tests. There is a noticeable peak around 17-18 marks, indicating that the majority of candidates have strong aptitude skills in these areas.

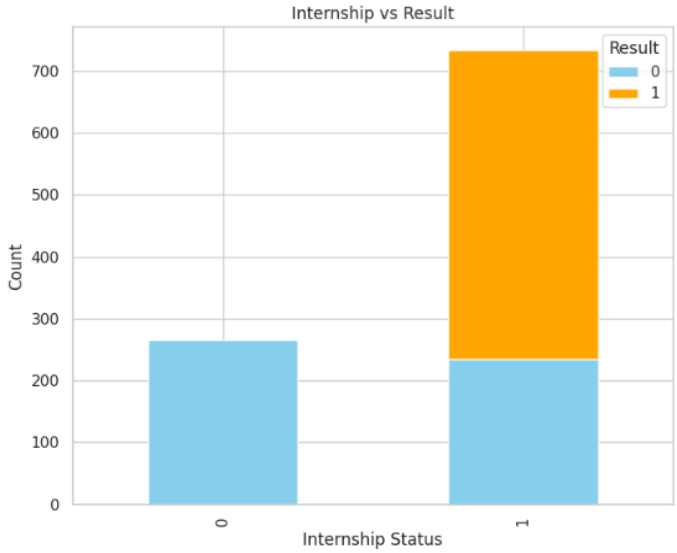


Fig 6 – Internship vs Result

From the above graph, it is evident that candidates who have completed internships i.e. Internship = 1 are more likely to be selected i.e. Result = 1. The majority of selected candidates have internship experience, while those without internships tend to have a higher proportion of non-selection. This highlights the importance of internship experience in improving hiring chances.

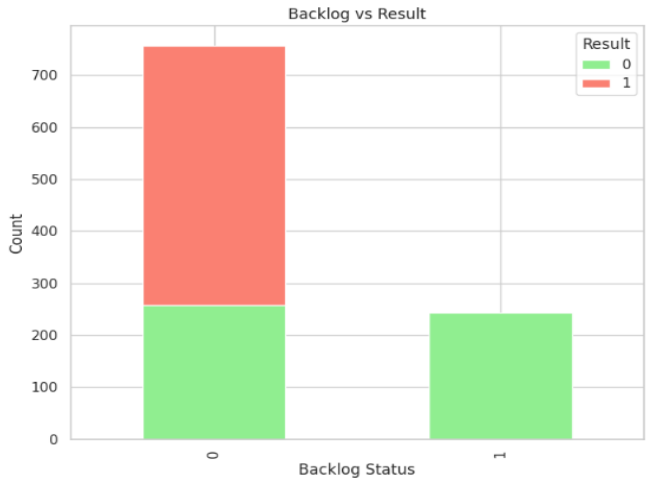


Fig 7 – Backlog vs Result

From the above graph, it shows that candidates with no backlogs i.e. Backlog = 0 have a higher chance of being selected i.e. Result = 1 compared to those with backlogs. Most candidates with backlogs tend to be not selected i.e. Result = 0. This suggests that having backlogs negatively impacts hiring outcomes.

Model Selection and Justification

For the hiring prediction application, the Random Forest Classifier was chosen as the primary machine learning model due to its strong predictive performance and robustness. It builds multiple decision trees and predicts by majority vote, reducing overfitting and handling both linear and non-linear data well.

The model is resilient to outliers and noise, and provides feature importance insights to help refine hiring decisions. Overall, it offers a good balance of performance and interpretability.

Model Training and Evaluation

After selecting the Random Forest Classifier, the model was trained using the preprocessed and normalized training dataset. During training, the algorithm constructed multiple decision trees using bootstrapped samples of the training data and combined their predictions through majority voting to form the final prediction. To evaluate the model's effectiveness, the trained model was tested on the unseen test data, and several performance metrics were calculated.

Accuracy was used as the primary metric to assess the overall correctness of the model, while precision, recall, and F1-score provided deeper insight into how well the model could correctly identify eligible and ineligible candidates, especially in cases where the dataset might be imbalanced. Additionally, a confusion matrix was analyzed to visualize the true positive, true negative, false positive and false negative predictions. To ensure the robustness of the model and to avoid overfitting, k-fold cross-validation was employed, where the data was partitioned into k subsets. The model was trained and validated k times, each time using a different subset as the validation set, ensuring the model's performance was consistent across different samples of data.

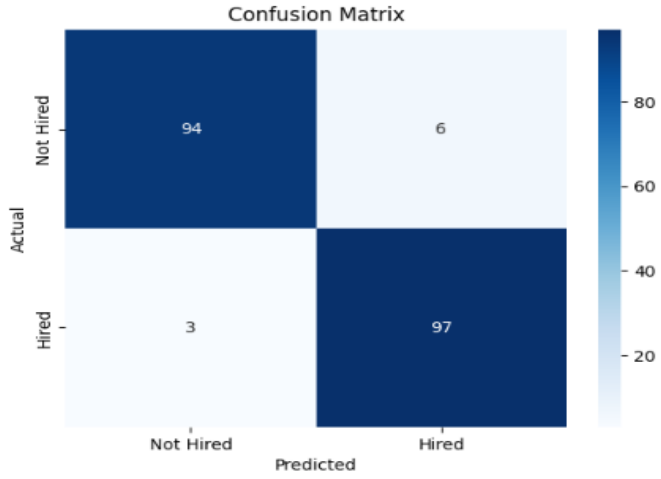


Fig 8 - Confusion Matrix

From the above graph, the confusion matrix indicates high classification performance. Out of the actual "Not Hired" candidates, 94 were correctly predicted as "Not Hired," while 6 were misclassified as "Hired." Similarly, out of the actual "Hired" candidates, 97 were correctly classified as "Hired," and only 3 were misclassified as "Not Hired." This demonstrates the model's strong predictive accuracy and its ability to correctly distinguish between hired and not-hired candidates, with minimal errors.

Hyperparameter Tuning

To improve the Random Forest Classifier's performance, Hyperparameter tuning was conducted by adjusting key parameters. The number of trees i.e. n_estimators, maximum tree depth i.e. max_depth, and minimum samples per leaf i.e. min_samples_leaf were optimized to balance accuracy and generalizability. Additionally, different split criteria, such as Gini impurity and entropy, were experimented with to evaluate their impact on tree construction. To identify the best combination of these hyperparameters, Grid Search and Randomized Search techniques were applied along with cross-validation, enabling the selection of parameter values that maximized the model's accuracy and stability.

Feature Importance Analysis

Understanding which features have the greatest influence on the hiring predictions is essential for transparency and continuous model improvement. Leveraging the Random Forest Classifier's ability to compute feature importance scores, the individual contributions of features such as academic scores, internship experience, backlog status, and aptitude test results were ranked.

This ranking provided valuable insights into which candidate attributes were most predictive of employability, enabling recruiters and administrators to better understand the decision-making process of the model. The analysis also guided candidates by highlighting the critical areas they should focus on to improve their chances of selection.

In addition, features ranked lower in this analysis were contemplated for elimination in subsequent versions of the model to make the model structure simpler, decrease computational cost, and lower noise in the data.

International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 13, Issue 5, May-2025, Available online at: www.ijaresm.com

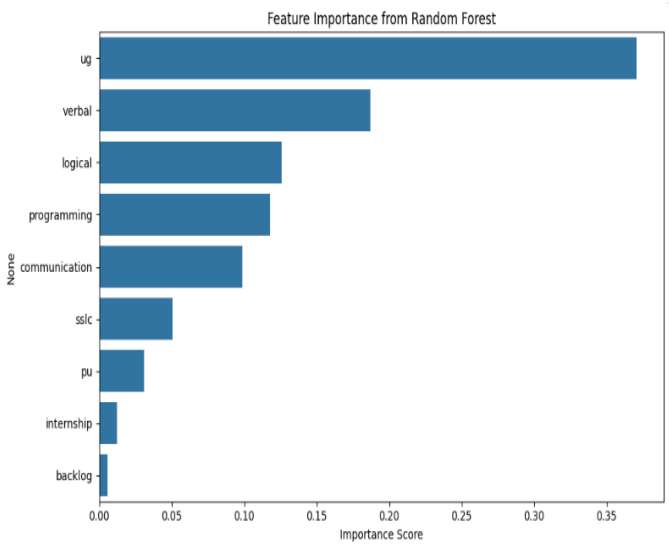


Fig 9 - Feature importance from Random forest

From the above graph, UG CGPAis the most important factor in predicting hiring decisions, followed by Verbal Skills,Logical Reasoning,andProgramming Skills. Communication Skills also played a significant role, highlighting the importance of both technical and soft skills. Features like SSLC, PU, and Internship Experience had a smaller impact, while Backlog Status was the least influential.

Model Validation and Testing

To ensure that the trained Random Forest model generalized well beyond the initial training dataset, strict validation and testing were conducted. The model was evaluated on a separate validation dataset or new candidate data, mimicking real-world scenarios to assess its predictive reliability and robustness. Statistical testing was performed to evaluate the consistency and stability of the model's predictions and to detect any potential bias or variance issues. Additionally, an error analysis was carried out by examining the incorrect predictions in detail to identify any common patterns, data anomalies, or feature-related problems. This analysis provided valuable feedback for further data cleaning, feature engineering, or even refining the model to improve future prediction accuracy.

Model Saving and Deployment

Once the Random Forest model was fully trained, optimized, and validated, preparation for deployment was undertaken. The trained model was serialized and saved using Python libraries such as joblib or pickle, enabling the model to be efficiently loaded and used for predictions without retraining each time. Integration testing was performed to incorporate the serialized model into the Flask backend of the hiring prediction application, ensuring seamless real-time prediction functionality based on user inputs. Additionally, performance optimization was addressed to minimize prediction latency, providing users with fast and responsive results when they submit their data. This step ensured that the model was production-ready and capable of delivering reliable predictions within the live application environment.

Technologies And Libraries Used

Component	Technology/Library
Web Framework	Flask
Data Handling	Pandas, NumPy
Machine Learning	Scikit-learn, Joblib, Matplotlib
Front-End	HTML and CSS
Database Systems	SQLite
Model Serialization	Pickle
Security	Bcrypt, Session Management

System Design

The Hiring Prediction Web Application is designed with a clear and efficient architecture that integrates machine learning and a Flask web application. This system is modular, with each component serving a specific function to streamline and automate the hiring process. The application is divided into two main blocks: the User Block and the Admin Block, each with distinct roles and functionalities.

International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 13, Issue 5, May-2025, Available online at: www.ijaresm.com

5.1 User Block Diagram

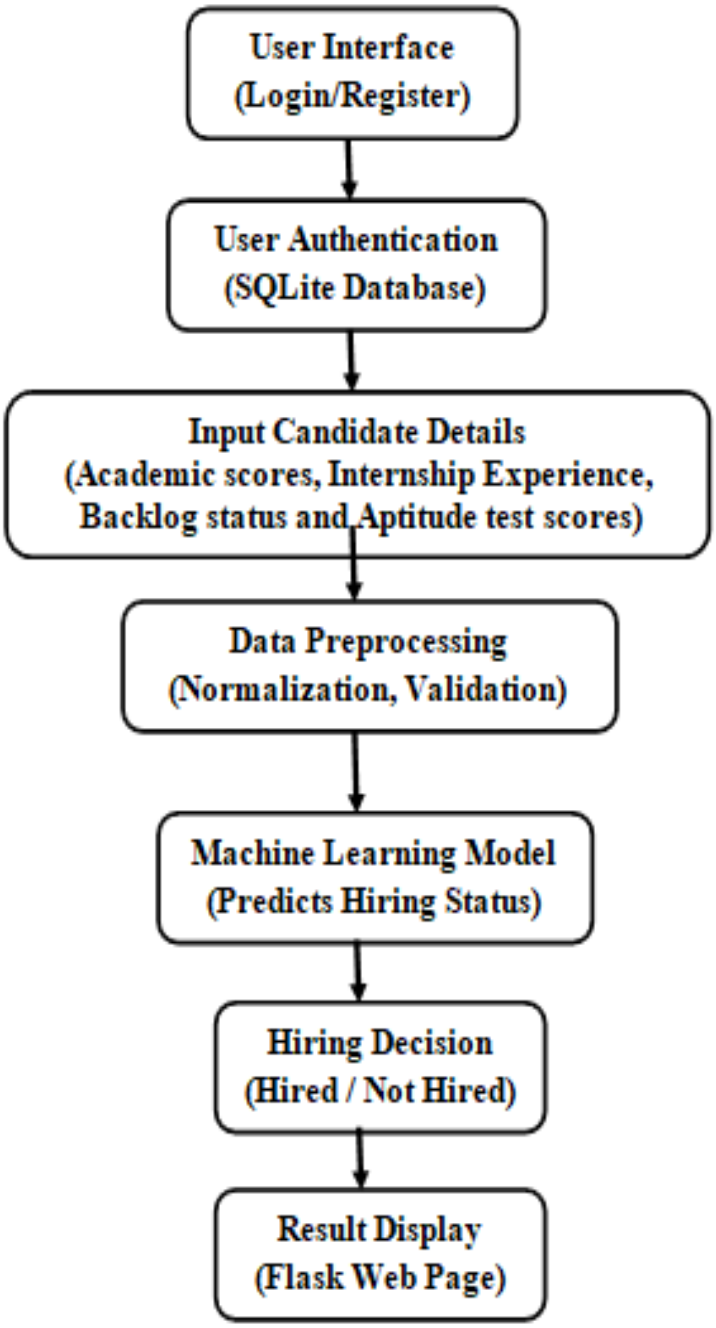


Fig 10 – User Block Diagram

User Block Diagram Explanation of Hiring Prediction Web Application

The Hiring Prediction Web Application follows a structured workflow, integrating user authentication, data processing, and machine learning-based predictions. Below is a detailed explanation of each component in the user block diagram.

1. User Interface (Login/Register Page)

The application starts with a login/register interface where users can create an account or log in if they already have one.
The frontend is built using HTML, and CSS, ensuring a user-friendly experience.
New users need to register by entering their username and password.

Once logged in, users can proceed to the prediction page.

2. User Authentication (SQLite Database)

The application uses SQLite for user authentication.
When a user logs in, their username and hashed password are verified against stored credentials.
If authentication is successful, they are redirected to the prediction page; otherwise, an error message is displayed.

3. Input Candidate Details (Academic scores, Internship Experience, Backlog status, and Aptitude test scores)

After login, users are prompted to input their academic scores, Internship experience, Backlog status, and Aptitude test scores in designated fields.
Input parameters include:
SSLC Percentage (0-100)
PU Percentage (0-100)
UG CGPA (0-10)
Logical Reasoning Score (0-20)
Verbal Skills Score (0-20)
Programming Skills Score (0-20)
Communication Skills Score (0-20)
Internship Experience (Yes/No)
Backlog Status (Yes/No)

4. Data Preprocessing (Validation and Normalization)

The application ensures that all input values are within valid ranges, for example SSLC and PU percentages are between 0-100.

Data is normalized using a scaler (joblib) to ensure that it matches the format expected by the trained model.

If the input is invalid, for example negative values or text instead of numbers, the system prompts the user to correct it.

5. Machine Learning Model (Predicts Hiring Status)

The trained machine learning model loaded using joblib is used to predict whether the candidate is eligible for hiring.
The hiring decision logic is as follows:
Condition 1: The candidate is hired if:
SSLC $\geq$ 50%, PU $\geq$ 35%, UG CGPA > 5.0
All Aptitude test scores like Logical, Verbal, Programming, Communication $\geq$ 15
Otherwise, the candidate is not hired.

6. Hiring Decision (Hired / Not Hired)

Based on the machine learning model output, the system determines whether the candidate meets the hiring criteria.
If eligible, the output will display "Congratulations! You are Hired "
If not eligible, the output will show "Sorry, you are Not Hired. Keep improving! "

7. Result Display (Flask Web Page)

The hiring status is displayed on the prediction page.
If the candidate is hired, a success message is shown.
If not hired, the system encourages them to improve their scores.
The user has an option to predict their history.

5.1.22 Administrators Block Diagram

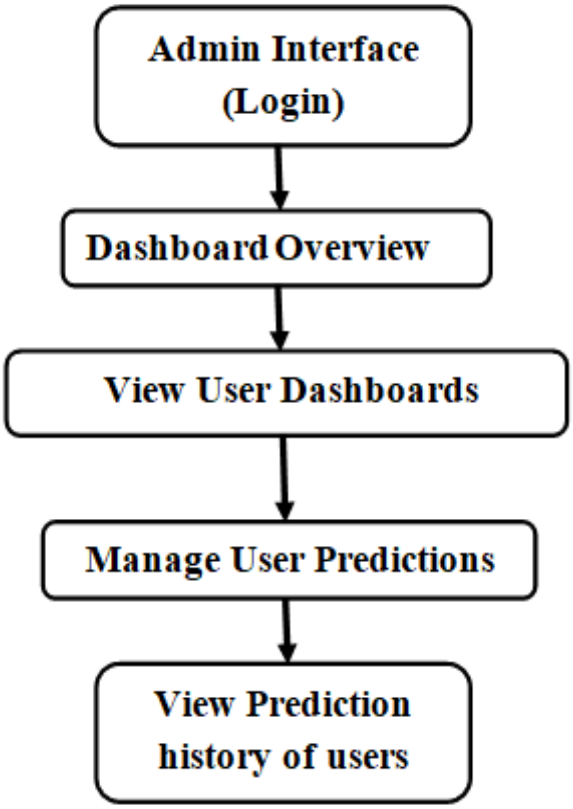


Fig 11 – Administrators Block Diagram

Administrators Block Diagram Explanation of Hiring Prediction Web Application

The admin block diagram for the Hiring Prediction Web Application provides a structured and efficient interface for administrators. It allows admins to manage users, oversee predictions, and control the machine learning model, ensuring effective application management.

1. Admin Interface (Login/Register)

Secure login with a username and password. Password hashing ensures data security. Session management prevents unauthorized access. Optional registration is available for multiple admins if enabled.

2. Dashboard Overview

Displays the number of registered users, number of predictions made, and model performance metrics like accuracy, precision, recall. Provides a visual representation of user engagement like active users, most frequent users and direct links to detailed user dashboards, prediction history, and model management.

3. View User Dashboards

Lists all registered users with search functionality. Displays user details like name, academic scores, aptitude test scores, internship and backlog status, prediction result, submitted at like date and time. Provides access to user-specific prediction history for monitoring user activities.

4. Manage User Predictions

Shows all predictions made by users in a tabular format. Allows admins to edit, delete, or update predictions. Provides filtering options by result like hired or not hired. Allows marking predictions as verified or reviewed.

5. View Prediction History of Users

Displays each prediction made by users, including user details likename, academic scores, aptitude test scores, internship and backlog status, prediction result, submitted at like date and time.

RESULT AND DISCUSSION

The Hiring Prediction Web Application provides accurate predictions based on candidate data like academic scores, internship experience, backlog status, and aptitude test scores successfully automates candidate evaluation, achieving high prediction accuracy. It reduces manual screening and speeds up the recruitment process. The application follows a structured workflow, allowing users to register, log in, enter academic details, take aptitude tests, submit their details for evaluation, users can view their prediction history, and admins also follows a structured workflow like to log in, view the user's prediction history. The machine learning model showed strong accuracy in evaluating eligibility. Input validation and security features like bcrypt improved system reliability. Overall, the system supports efficient and data-driven hiring decisions. It serves as a foundation for future innovations in data-driven human resource management.

6.1 Final Outputs of Regular Users:

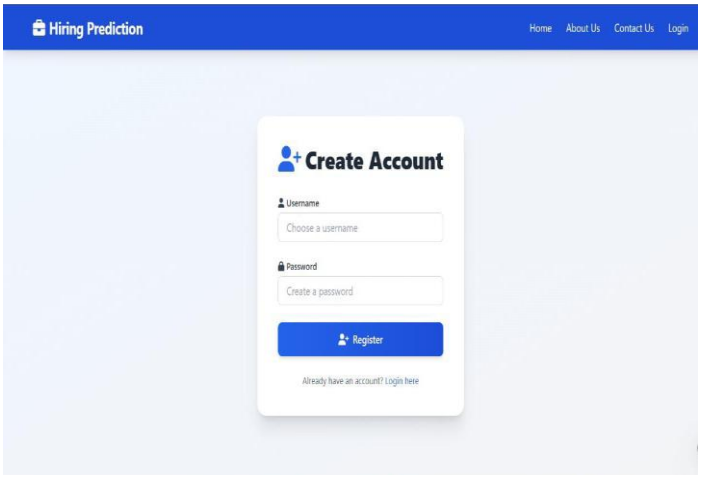


Fig 12 - Register Page

In the above image, if the user is new to the application then the user has to register with username and password.

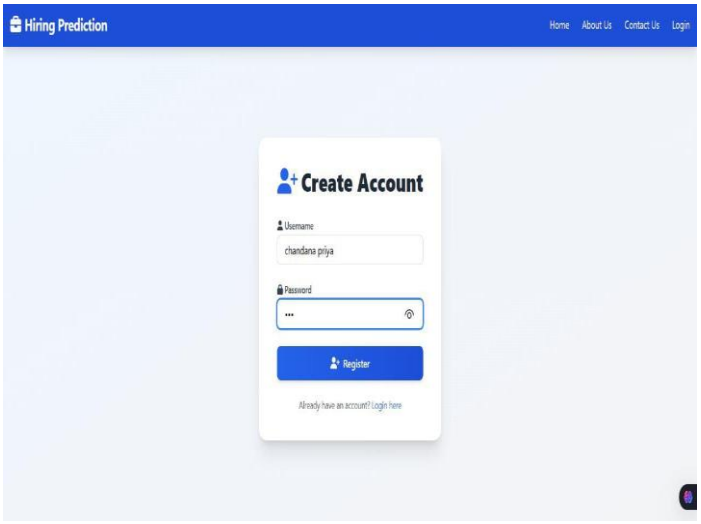


Fig 13 - Register Page with details

In the above image, the user created an account by entering the username and password.

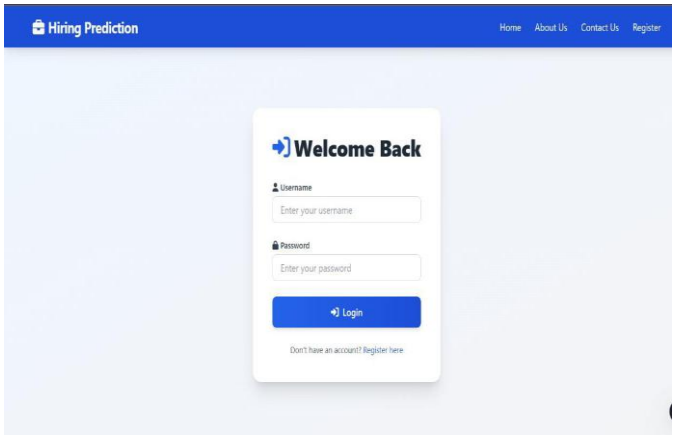


Fig 14 - Login Page

In the above image, the user should log in with a username and Password after registration.

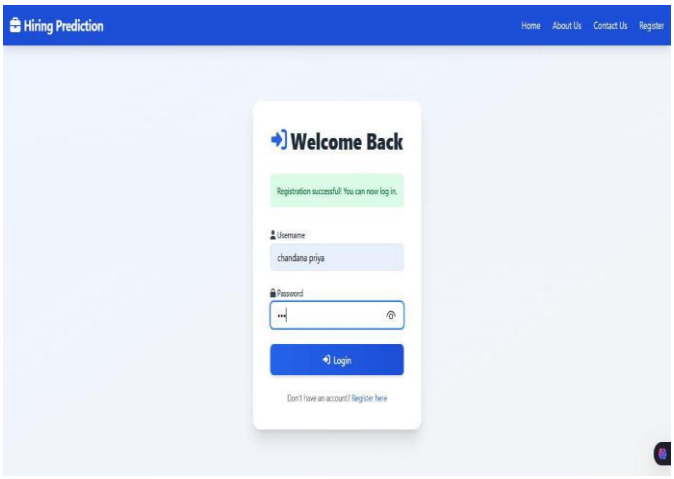


Fig 15 - Login Page with details.

In the above image, the user login by entering username and Password after registration.

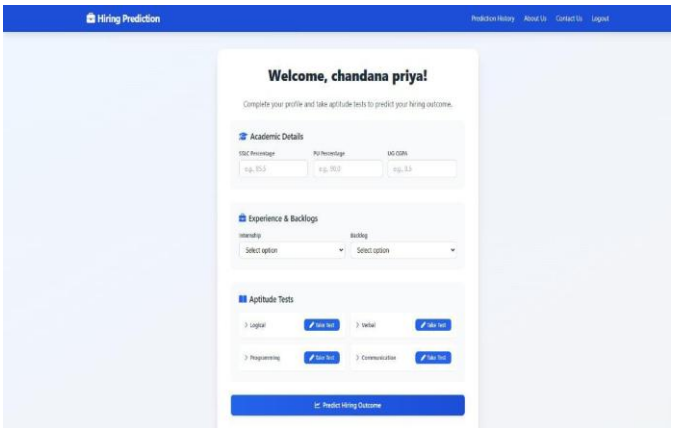


Fig 16– Home Page

In the above image, the message was shown like “Welcome, Chandana Priya”. Next, we should enter the scores of all candidate details and click on the take test button to take test then we have to click the predict button.

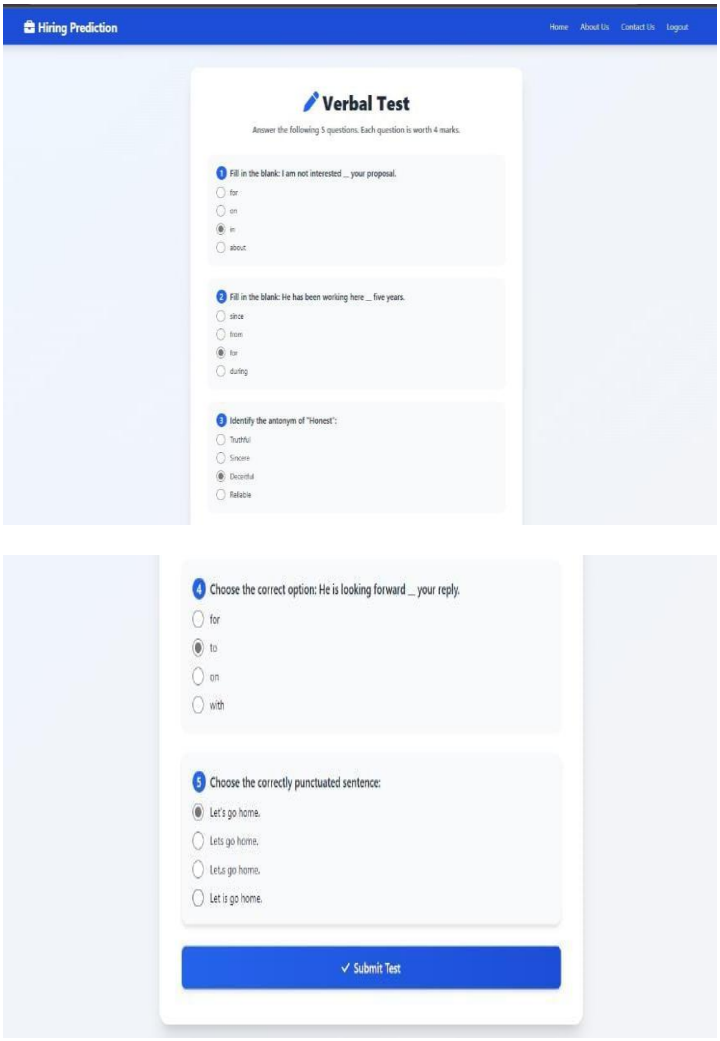
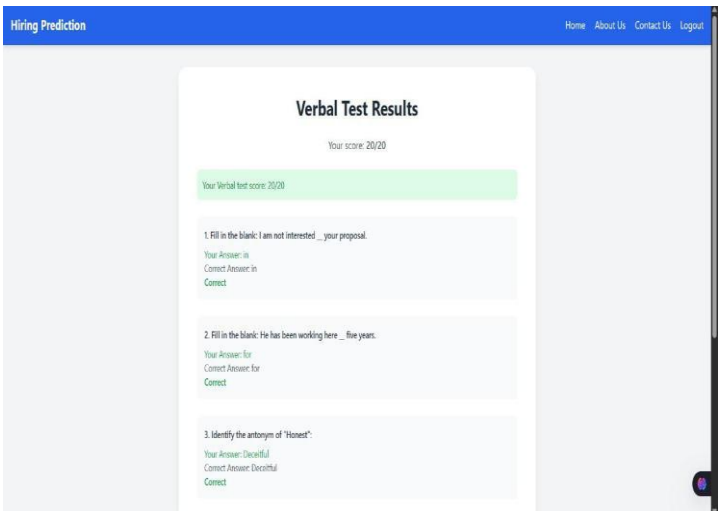


Fig 17 – Test Page

In the above image, user should select the take test button and select the options for each question. It is same as for all Communication, Logical, and Programming. After test, submit the test.



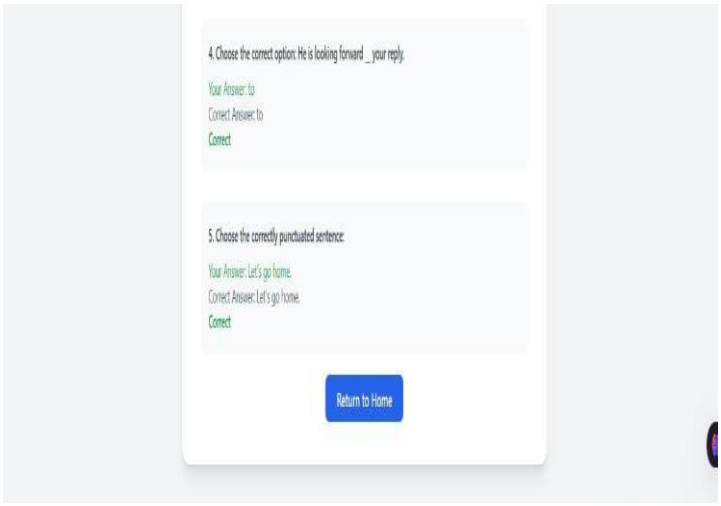

4. Choose the correct option: He is looking forward _ your reply.
Your Answer: to
Correct Answer: to
Correct
5. Choose the correctly punctuated sentence:
Your Answer: Let's go home.
Correct Answer: Let's go home.
Correct
Return to Home

Fig 18 – Test Result Page

In the above image, test result is displayed after submitting test for each question. It is same as for all Communication, Logical, and Programming. Next, Return to home button has to select.

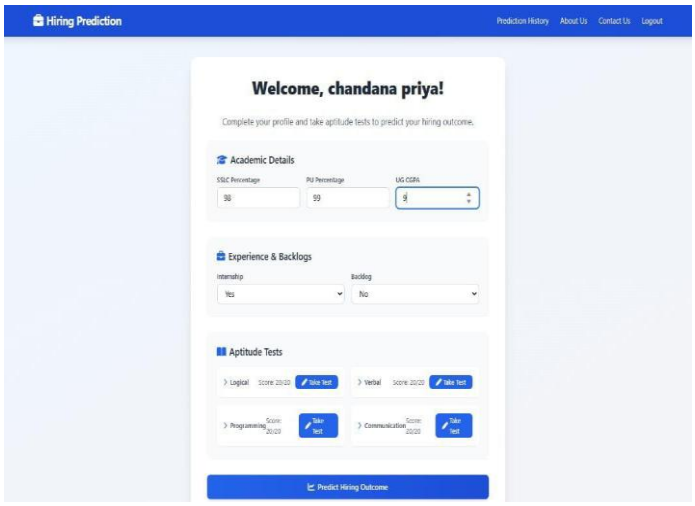


Fig 19– Home Page with details

In the above image, after test completion, automatically scores will be displayed. Next we should enter the academic scores and select the internship and backlog with yes or no options, then click on the button “Predict Hiring Outcome”.

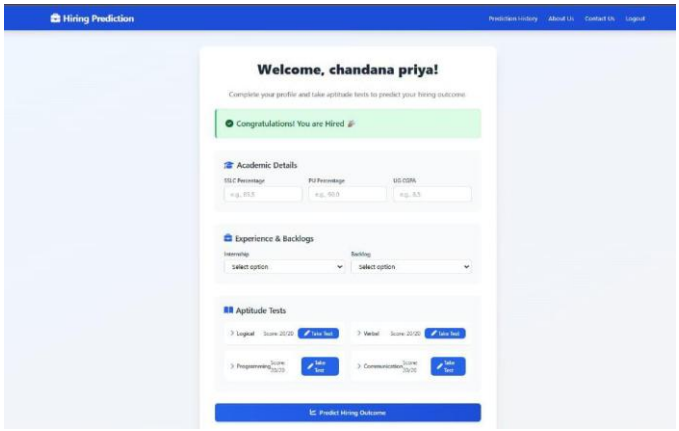


Fig 20 – Result Page – Hired

In the above image, after submitting the details, it will predict “Congratulations! You are hired” based on the eligibility.

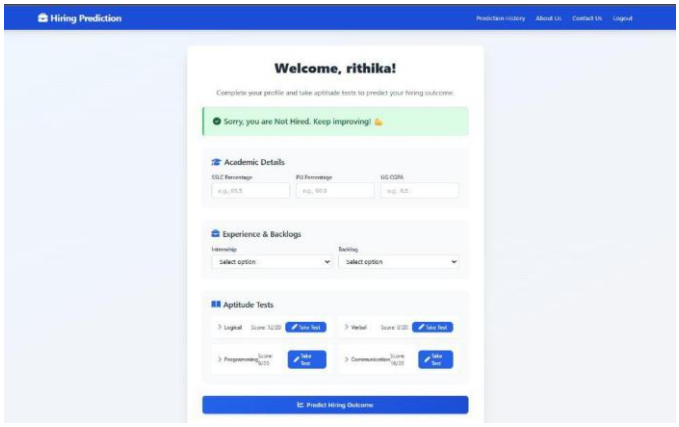


Fig 21 – Result Page – Not Hired

In the above image, after submitting the details, it will predict “Sorry, you are Not Hired, Keep improving!” based on the eligibility.

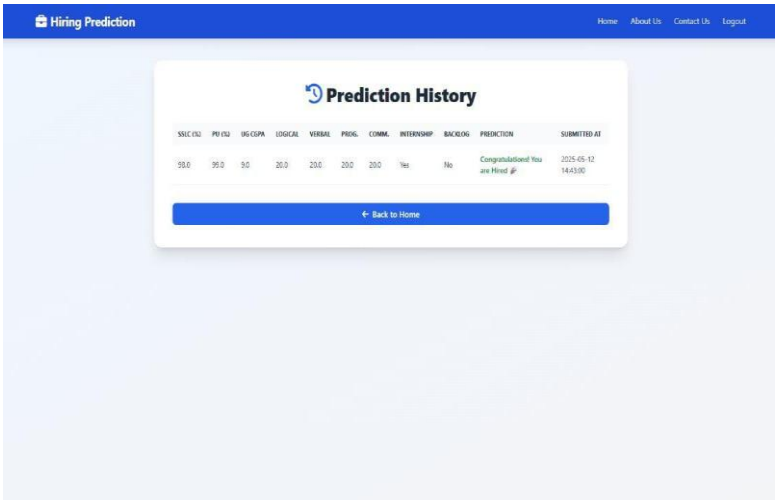
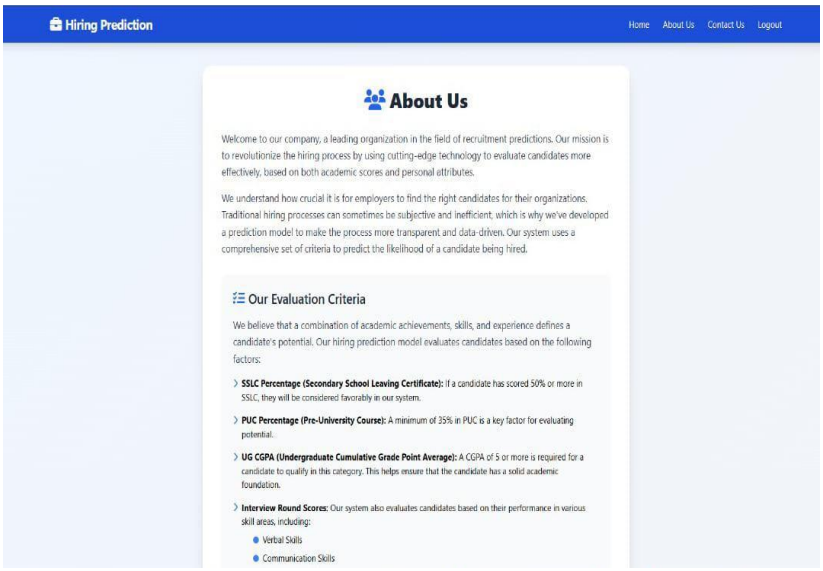


Fig 22 – Prediction history Page

In the above image, after getting the eligibility, users can view their prediction history with date and time.


Hiring Prediction
Home
About Us
Contact Us
Logout
About Us
Welcome to our company, a leading organization in the field of recruitment predictions. Our mission is to revolutionize the hiring process by using cutting-edge technology to evaluate candidates more effectively, based on both academic scores and personal attributes.
We understand how crucial it is for employers to find the right candidates for their organizations. Traditional hiring processes can sometimes be subjective and inefficient, which is why we've developed a prediction model to make the process more transparent and data-driven. Our system uses a comprehensive set of criteria to predict the likelihood of a candidate being hired.
Our Evaluation Criteria
We believe that a combination of academic achievements, skills, and experience defines a candidate's potential. Our hiring prediction model evaluates candidates based on the following factors:
SSLC Percentage (Secondary School Leaving Certificate): If a candidate has scored 50% or more in SSLC, they will be considered favorably in our system.
PUC Percentage (Pre-University Course): A minimum of 35% in PUC is a key factor for evaluating potential.
UG CGPA (Undergraduate Cumulative Grade Point Average): A CGPA of 5 or more is required for a candidate to qualify in this category. This helps ensure that the candidate has a solid academic foundation.
Interview Round Scores: Our system also evaluates candidates based on their performance in various skill areas, including:
Verbal Skills
Communication Skills

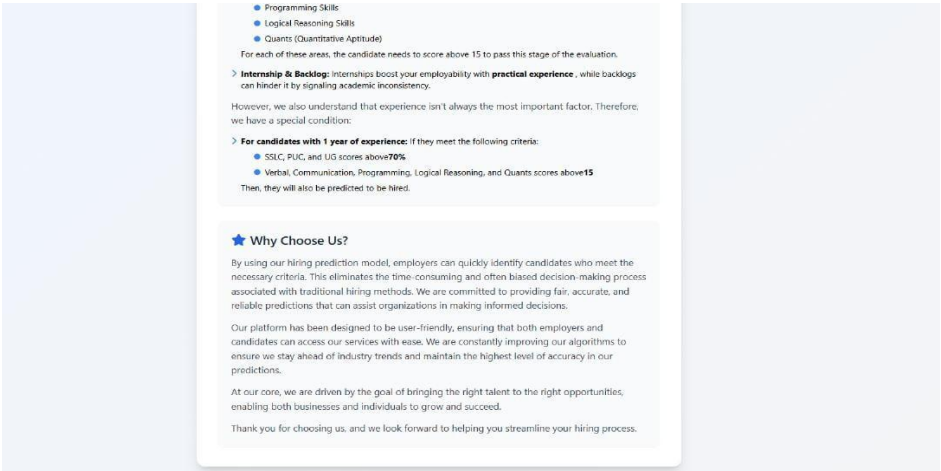

Programming Skills
Logical Reasoning Skills
Quants (Quantitative Aptitude)
For each of these areas, the candidate needs to score above 15 to pass this stage of the evaluation.
Internship & Backlog: Internships boost your employability with practical experience , while backlogs can hinder it by signaling academic inconsistency.
However, we also understand that experience isn't always the most important factor. Therefore, we have a special condition:
For candidates with 1 year of experience: If they meet the following criteria:
SSLC, PUC, and UG scores above70%
Verbal, Communication, Programming, Logical Reasoning, and Quants scores above15
Then, they will also be predicted to be hired.
Why Choose Us?
By using our hiring prediction model, employers can quickly identify candidates who meet the necessary criteria. This eliminates the time-consuming and often biased decision-making process associated with traditional hiring methods. We are committed to providing fair, accurate, and reliable predictions that can assist organizations in making informed decisions.
Our platform has been designed to be user-friendly, ensuring that both employers and candidates can access our services with ease. We are constantly improving our algorithms to ensure we stay ahead of industry trends and maintain the highest level of accuracy in our predictions.
At our core, we are driven by the goal of bringing the right talent to the right opportunities, enabling both businesses and individuals to grow and succeed.
Thank you for choosing us, and we look forward to helping you streamline your hiring process.

Fig 23 – About Us Page

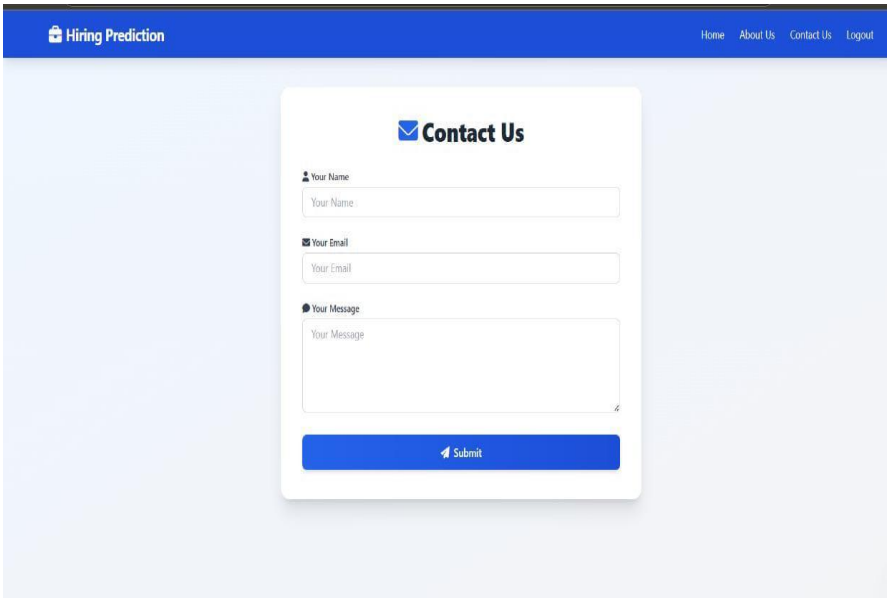


Fig 24 – Contact Us Page

6.2 Final Outputs of Administrators:

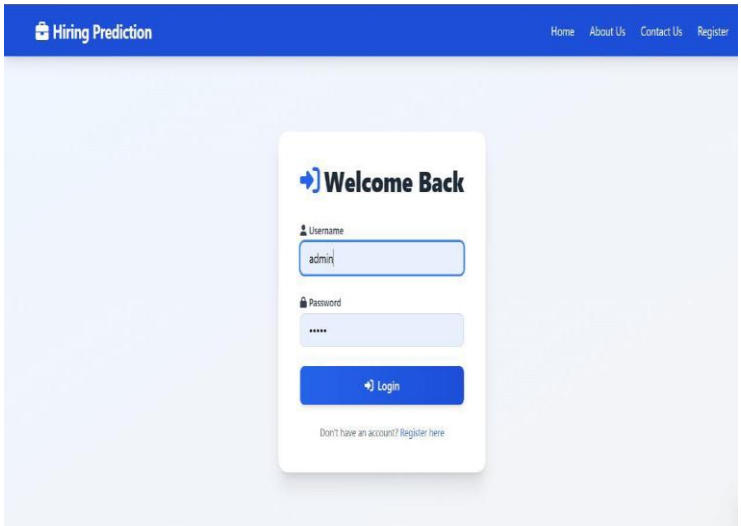


Fig 25 –Login Page

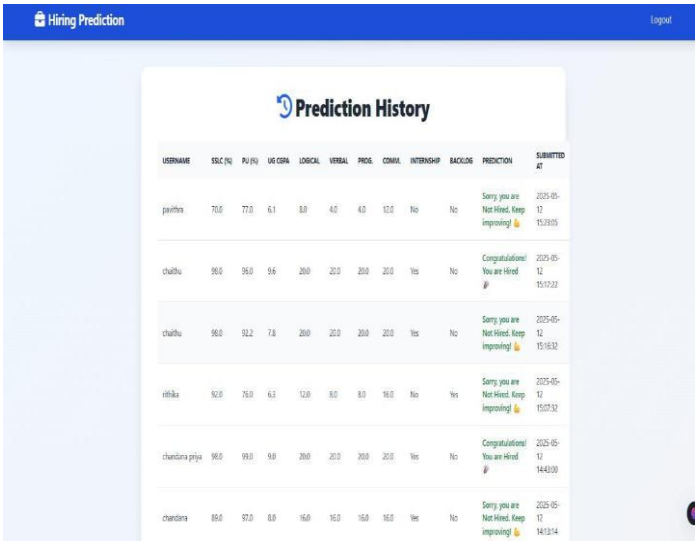


Fig 26 –Prediction history Page

In the above image, admin can view the user's prediction history with username, academic scores, internship and backlog status, prediction, submitted date and time.

CONCLUSION

The Hiring Prediction Web Application showcases the power of machine learning to transform the recruitment landscape by providing users with a data-driven way to understand their hiring potential. By considering important features such as SSLC Percentage, PUC Percentage, UG CGPA, Logical Reasoning, Verbal Skills, Programming Skills, Communication Skills, Internship Experience, and Backlog Status, the system offers a well-rounded evaluation of candidates. Users benefit from transparent and consistent feedback generated through a predefined hiring logic, which helps them recognize their strengths and areas that require improvement. This encourages better preparation and informed decision-making, empowering candidates to take control of their career paths.

For administrators, the application streamlines and automates the recruitment process by utilizing the Random Forest classifier, known for its high accuracy in handling complex datasets and capturing nuanced patterns. This automation reduces manual effort and biases, ensuring fairer and more objective candidate assessments. The integration of Flask and a secure SQLite database provides a reliable backend infrastructure for managing candidate information efficiently. Administrators can monitor hiring trends, generate reports, and make data-backed decisions that improve organizational outcomes. The system's robust validation mechanisms ensure data integrity and user security throughout the recruitment lifecycle.

The use of feature engineering techniques such as normalization and encoding further improves the prediction accuracy of the Random Forest model, making it well-suited for real-world hiring challenges. The application supports real-time predictions, enabling both users and administrators to receive immediate insights that aid timely decision-making. This project bridges the gap between traditional hiring inefficiencies and modern technological advancements. It empowers users with clear, actionable insights while equipping administrators with an efficient and unbiased recruitment tool. The Hiring Prediction Web Application marks a significant step toward a more objective, transparent, and efficient hiring process. Its adaptability and potential for growth make it a valuable asset for organizations aiming to optimize recruitment outcomes while saving time and resources. This initiative demonstrates how machine learning can be practically applied to enhance fairness and productivity in hiring, benefiting both candidates and companies alike.

REFERENCES

[1]. L. Yang, et al., "Database Management and Performance Analysis for Student Placement using ML", Proceedings of the ACM Conference on Computing and Data Analytics, 2018.
[2]. Alfiani, M., et al., "Mapping Student Performance Patterns Using Data Mining," Elsevier Procedia Computer Science, vol. 157, pp. 221–228, 2019.
[3]. A. Meena and M. Gopal, "Recruitment Prediction System Using Logistic Regression and Decision Tree," Springer Lecture Notes on Data Engineering and Communications Technologies, vol. 34, pp. 456–463, 2020.
[4]. T. Rajesh and G. Swetha, "Employee Hiring Using Machine Learning," International Journal of Engineering Research & Technology (IJERT), vol. 10, no. 8, pp. 101–104, 2021.

[5]. K. Arvind and J. Sneha, "Predictive Analytics for Hiring using SVM and KNN," 2022 IEEE International Conference on Smart Computing and Communication (ICSCC), pp. 312–317, doi: 10.1109/ICSCC.2022.00123.
[6]. S. M. B. Sheetal and S. Bakare, "Campus Placement Prediction Using Machine Learning Techniques," Springer Advances in Intelligent Systems and Computing, vol. 1375, pp. 451–460, 2022.
[7]. R. Sumitha, et al., "Predicting Student Placement Outcomes Using Decision Trees," 2021 IEEE International Conference on Computational Intelligence and Data Science (ICCIDS), pp. 78–83, doi: 10.1109/ICCIDS.2021.00256.
[8]. S. Elayidom, S. M. Idikkula, J. Alexander, and A. Ojha, "Applying Data Mining Techniques for Placement Chance Prediction," 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies (ACT), pp. 669–671, 2009, doi: 10.1109/ACT.2009.169.
[9]. J. Nagaria and S. V. S., "Utilizing Exploratory Data Analysis for the Prediction of Campus Placement for Educational Institutions," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–7, 2020, doi: 10.1109/ICCCNT49239.2020.9225441.
[10]. S. Venkatachalam, "Data Mining Classification and Analytical Model of Prediction for Job Placements Using Fuzzy Logic," 2021 IEEE International Conference on Trends in Electronics and Informatics (ICOEI), pp. 342–347, 2021.
[11]. P. Manvitha and N. Swaroopa, "Campus Placement Prediction Using Supervised Machine Learning Techniques," International Journal of Applied Engineering Research, vol. 14, no. 9, pp. 2188–2193, 2019.
[12]. A. Patel and R. Shah, "Predicting Student Placements Using Classification Algorithms," International Journal of Computer Applications, vol. 182, no. 31, pp. 12–17, 2018, doi: 10.5120/ijca2018918210.
[13]. M. Ruparel and P. Swaminarayan, "Enhancing Student Placement Predictions with Advanced Machine Learning Techniques," Journal of Information Systems Engineering and Management, vol. 10, no. 1s, 2025.
[14]. R. Mkhinini Gahar, A. Hidri, and M. Sassi Hidri, "Let's Predict Who Will Move to a New Job," arXiv preprint arXiv: 2309.08333, 2023.
[15]. P. Thakar, A. Mehta, and Manisha, "Unified Prediction Model for Employability in Indian Higher Education System," arXiv preprint arXiv: 2407.17591, 2024.
[16]. T. Chakraborty, "Imbalanced Ensemble Classifier for Learning from Imbalanced Business School Data Set," arXiv preprint arXiv: 1805.12381, 2018.
[17]. C. Kumbhar and S. S. Sridhar, "Trend Analysis of University Placement by Using Machine Learning Algorithms," International Journal of Engineering & Technology, vol. 7, no. 2.4, pp. 178–181, 2018.
[18]. S. R. Khonde et al., "Pre-Placement Prediction System Using Machine Learning," International Journal of Advanced Research in Computer and Communication Engineering, vol. 11, no. 12, 2022.
[19]. M. Gupta and S. D. Pandya, "Machine Learning-Based Predictive Modeling for Recruitment and Selection in Campus Placement Processes," Journal of Electrical Systems, vol. 20, no. 10s, 2024.
[20]. R. C. K. et al., "Identifying Patterns and Trends in Campus Placement Data Using Machine Learning," International Journal of Education and Management Engineering, vol. 15, no. 1, pp. 10–24, 2025.
[21]. K. Eswara Rao et al., "An Advanced Machine Learning Approach for Student Placement Prediction and Analysis," International Journal of Performability Engineering, vol. 19, no. 8, pp. 536–546, 2023.
[22]. M. S. Hassan and A. Goyal, "Predicting Student Placements Using Machine Learning Algorithms," Tuijin Jishu/Journal of Propulsion Technology, vol. 44, no. 4, 2023.